TCP Timers

• Operations that involve the TCP timers:
  – Lost packet retransmission
    • client and server side
  – Repeating the slow-start phase
    • usually only client side
  – Reclaiming state after connection termination
    • usually only server side
Retransmission Timer

- TCP Connection Delay
  - 3-way handshake

- Web Transfer Delay
  - Mostly caused by congestion window size
Slow-start Restart

• How does this affect persistent connections?
• Ways to reduce the restart penalty
  – Disable timer
  – Make timeout longer
  – Dynamic congestion window
  – Pace transmission packets
TIME_WAIT State

- Why is this state needed?
- Effect on web servers
- Reducing overhead
  - Change TCP
  - Change HTTP
HTTP/TCP Layering

- Some issues that can apply to both the HTTP and the TCP layers:
  - Aborted HTTP Transfers
  - Nagle’s algorithm
  - Delayed Acknowledgements
Aborted HTTP Transfers

- HTTP has no support for aborts
- How does a user’s abort command impact web performance?

Details of TCP connection abort:
- FIN packet abort
- RST packet abort
Nagle’s Algorithm

- Made for older protocols
- Reduces amount of small packets
- Usability with persistent connections
- Why not just disable it?
Delayed Acknowledgements

- Why delay ACK packets?
- Interaction with HTTP traffic
Multiplexing TCP Connections

- What is good about using more than one connection for the web?
- Images with webpages
- Proxies
- Fooling the congestion window
- Problems with multiple connections?
- Fairness
- Server load
- Latency
Server Overheads

- System call overhead
  - Multiple system calls for single operations
  - Some can be combined:
    - GET requests
    - Send Header & Body with one call
    - Send Response and FIN with one call

- Limit simultaneous connections

- Close persistent connections
Any Questions?